

# ????? ??????????????????

Подробнее (Оригинальная статья): [здесь](#)

## ????????? Easy-RSA

```
sudo apt update
sudo apt install easy-rsa
```

????????????? ?????????????? ??? ?????????????????????? ?????????? ??????????

```
mkdir ~/easy-rsa
```

Создайте символические ссылки с помощью команды `ln`:

```
ln -s /usr/share/easy-rsa/* ~/easy-rsa/
```

Чтобы ограничить доступ к созданной директории PKI, используйте команду `chmod` для предоставления доступа к ней только владельцу:

```
chmod 700 /home/sammy/easy-rsa
```

Затем инициализируйте PKI в директории `easy-rsa`:

```
cd ~/easy-rsa
./easyrsa init-pki
```

### Output

```
init-pki complete; you may now create a CA or requests.
Your newly created PKI dir is: /home/sammy/easy-rsa/pki
```

????????? ?????????? ??????????????????

```
cd ~/easy-rsa
nano vars
```

Открыв файл, вставьте следующие строки и измените каждое выделенное значение для отражения информации о вашей организации. При этом важно, чтобы ни одно значение не оставалось пустым:

~/easy-rsa/vars

```
set_var EASYRSA_REQ_COUNTRY    "US"  
set_var EASYRSA_REQ_PROVINCE   "NewYork"  
set_var EASYRSA_REQ_CITY       "New York City"  
set_var EASYRSA_REQ_ORG        "DigitalOcean"  
set_var EASYRSA_REQ_EMAIL      "admin@example.com"  
set_var EASYRSA_REQ_OU         "Community"  
set_var EASYRSA_ALGO           "ec"  
set_var EASYRSA_DIGEST         "sha512"
```

Для создания корневой пары открытого и закрытого ключей для Центра сертификации необходимо запустить команду `./easy-rsa` еще раз, но уже с опцией `build-ca`:

```
./easysrsa build-ca
```

## Output

```
. . .  
Enter New CA Key Passphrase:  
Re-Enter New CA Key Passphrase:  
. . .  
Common Name (eg: your user, host, or server name) [Easy-RSA CA]:  
  
CA creation complete and you may now import and sign cert requests.  
Your new CA certificate file for publishing is at:  
/home/sammy/easy-rsa/pki/ca.crt
```

Если вы не хотите вводить пароль при каждом взаимодействии с ЦС, вы можете запустить команду `build-ca` с опцией `nopass`:

```
./easysrsa build-ca nopass
```

????????????????? ??????????? ?????????????? ?????? ???????????????

Чтобы импортировать публичный сертификат ЦС во вторую систему Linux, например на сервер или локальный компьютер, нужно предварительно получить копию файла `ca.crt` с сервера ЦС. Вы можете использовать команду `cat` для ее вывода в терминал, а затем скопировать и вставить ее в файл на втором компьютере, который импортирует сертификат. Также вы можете использовать `scp`, `rsync` и другие подобные инструменты для передачи файла между системами. Мы используем для копирования и вставки текстовый редактор `nano`, поскольку этот вариант подойдет для всех систем.

Запустите следующую команду на сервере ЦС от имени пользователя без прав root:

```
cat ~/easy-rsa/pki/ca.crt
```

На терминале появится примерно следующее:

Output

```
-----BEGIN CERTIFICATE-----  
MIIDSzCCAj0gAwIBAgIUcr9Crsv3FBEujrPZnZnU4nSb5TMwDQYJKoZIhvcNAQEL  
BQAwFjEUMBIGA1UEAwwLRWFzeS1SU0EgQ0EwHhcNMjAwMzE4MDMxNjI2WhcNMzAw  
...  
...  
-----END CERTIFICATE-----
```

Скопируйте все, включая строки `-----BEGIN CERTIFICATE-----` и `-----END CERTIFICATE-----` и символы дефиса.

Используйте `nano` или предпочитаемый текстовый редактор на второй системе Linux, чтобы открыть файл с именем `/tmp/ca.crt`:

```
nano /tmp/ca.crt
```

Теперь у нас имеется копия файла `ca.crt` на второй системе Linux, и мы можем импортировать сертификат в хранилище сертификатов операционной системы.

На системах с Ubuntu и Debian выполните следующие команды в качестве вашего пользователя без прав root для импорта сертификата:

```
sudo cp /tmp/ca.crt /usr/local/share/ca-certificates/  
sudo update-ca-certificates
```

Чтобы импортировать сертификат сервера ЦС в систему на базе CentOS, Fedora или RedHat, скопируйте и вставьте содержимое файла в файл `/tmp/ca.crt`, как описано в предыдущем примере. Затем скопируйте сертификат в директорию `/etc/pki/ca-trust/source/anchors/` и запустите команду `update-ca-trust`.

```
sudo cp /tmp/ca.crt /etc/pki/ca-trust/source/anchors/  
sudo update-ca-trust
```

Теперь вторая система Linux будет доверять любому сертификату, подписанному нашим сервером ЦС.

Если вы используете ЦС с веб-серверами и браузер Firefox, вам нужно будет импортировать публичный сертификат `ca.crt` в Firefox напрямую. Firefox не использует локальное хранилище сертификатов операционной системы. Подробную информацию о добавлении сертификата ЦС в Firefox можно найти в статье поддержки Mozilla [«Настройка Центров сертификации \(ЦС\) в Firefox»](#).

Если вы используете ЦС для интеграции со средой Windows или настольными компьютерами, ознакомьтесь с документацией по использованию `certutil.exe` для [установки сертификата ЦС](#).

????????? ?????????? ?? ?????????????? ??????????????? ? ??????  
?????????????????

????????? ? ?????????????? ?????????? ?????????? ???????????????

`openssl` обычно устанавливается по умолчанию в большинстве дистрибутивов Linux, но для уверенности стоит запустить в системе следующую команду:

```
sudo apt update
sudo apt install openssl
```

В первую очередь для создания CSR необходимо сгенерировать закрытый ключ. Чтобы создать закрытый ключ с помощью `openssl`, создайте директорию `practice-csr` и сгенерируйте ключ в этой директории. Мы будем выполнять этот запрос на фиктивном сервере под названием `sammy-server`, в отличие от случая создания сертификата для идентификации пользователя или другого ЦС.

```
mkdir ~/practice-csr
cd ~/practice-csr
openssl genrsa -out sammy-server.key
```

### Output

```
Generating RSA private key, 2048 bit long modulus (2 primes)
. . .
. . .
e is 65537 (0x010001)
```

Теперь у нас имеется закрытый ключ, с помощью которого можно создать CSR, используя утилиту `openssl`. Вам будет предложено заполнить ряд полей, в том числе указать страну, область и город. Вы можете ввести `.`, если хотите оставить поле пустым, но для реальных CSR лучше использовать правильные значения при указании своего расположения и

организации:

```
openssl req -new -key sammy-server.key -out sammy-server.req
```

## Output

```
. . .
-----
Country Name (2 letter code) [XX]:US
State or Province Name (full name) []:New York
Locality Name (eg, city) [Default City]:New York City
Organization Name (eg, company) [Default Company Ltd]:DigitalOcean
Organizational Unit Name (eg, section) []:Community
Common Name (eg, your name or your server's hostname) []:sammy-server
Email Address []:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
```

Когда вас устроит тема запроса тренировочного сертификата, скопируйте файл `sammy-server.req` на сервер ЦС с помощью `scp`:

```
scp sammy-server.req sammy@your_ca_server_ip:/tmp/sammy-server.req
```

## ?????????? CSR

Первым шагом для подписания вымышленного CSR будет импорт запроса сертификата с помощью скрипта `easy-rsa`:

```
cd ~/easy-rsa
./easymrsa import-req /tmp/sammy-server.req sammy-server
```

## Output

```
. . .
The request has been successfully imported with a short name of: sammy-server
You may now use this name to perform signing operations on this request.
```

Теперь вы можете подписать запрос, запустив скрипт `easymrsa` с опцией `sign-req`, указав затем тип запроса и общее имя, включаемое в CSR. Запрос может иметь тип `client`, `server` или `ca`. Поскольку мы тренируемся с сертификатом для вымышленного сервера, нужно использовать тип запроса `server`:

```
./easyrsa sign-req server sammy-server
```

В результатах вам будет предложено подтвердить, что запрос поступил из надежного источника. Для подтверждения введите `yes` и нажмите `ENTER`:

## Output

```
You are about to sign the following certificate.
Please check over the details shown below for accuracy. Note that this request
has not been cryptographically verified. Please be sure it came from a trusted
source or that you have verified the request checksum with the sender.

Request subject, to be signed as a server certificate for 3650 days:

subject=
  commonName          = sammy-server

Type the word 'yes' to continue, or any other input to abort.
Confirm request details: yes
. . .
Certificate created at: /home/sammy/easy-rsa/pki/issued/sammy-server.crt
```

Если вы зашифровали ключ ЦС, вам будет предложено ввести пароль.

Выполнив эти шаги, мы подписали CSR `sammy-server.req` с помощью закрытого ключа сервера ЦС в директории `/home/sammy/easy-rsa/pki/private/ca.key`. Полученный файл `sammy-server.crt` содержит открытый ключ шифрования тренировочного сервера, а также новую подпись от сервера ЦС. Подпись сообщает всем, кто доверяет ЦС, что они также могут доверять сертификату `sammy-server`.

Если бы это был запрос веб-сервера, сервера VPN или другого реального сервера, последним шагом на сервере ЦС стало бы распространение новых файлов `sammy-server.crt` и `ca.crt` с сервера ЦС на удаленный сервер, отправивший запрос CSR:

```
scp pki/issued/sammy-server.crt sammy@your_server_ip:/tmp
scp pki/ca.crt sammy@your_server_ip:/tmp
```

????? ??????????????

Иногда сертификат требуется отозвать, чтобы пользователь или сервер не могли его использовать. Например, это может потребоваться в случае кражи ноутбука, взлома веб-сервера, увольнения сотрудника, расторжения договора с подрядчиком и т. д.

Далее кратко описана процедура отзыва сертификата:

1. Для отзыва сертификата используется команда `./easysrsa revoke client_name`.
2. Сгенерируйте новый CRL с помощью команды `./easysrsa gen-crl`.
3. Переместите обновленный файл `crl.pem` на сервер или серверы, использующие ваш ЦС, а на этих системах скопируйте этот файл в директорию или директории программ, которые на него ссылаются.
4. Перезапустите все службы, использующие ваш ЦС и файл CRL.

С помощью этой процедуры вы можете отозвать любые сертификаты, которые ранее выпустили для вашего сервера. В следующих разделах мы подробно рассмотрим каждый шаг, начиная с команды `revoke`.

Для отзыва сертификата перейдите в директорию `easy-rsa` на вашем сервере ЦС:

```
cd ~/easy-rsa
```

Затем запустите скрипт `easysrsa` с опцией `revoke`, указав имя клиента, у которого хотите отозвать сертификат: В соответствии с приведенным выше практическим примером, сертификат имеет обычное имя `sammy-server`:

```
./easysrsa revoke sammy-server
```

Система предложит вам подтвердить отзыв сертификата. Введите `yes`:

Output

Please confirm you wish to revoke the certificate with the following subject:

subject=

commonName = sammy-server

Type the word 'yes' to continue, or any other input to abort.

Continue with revocation: yes

. . .

Revoking Certificate 8348B3F146A765581946040D5C4D590A

. . .

Обратите внимание на выделенное значение в строке `Revoking Certificate`. Это значение представляет собой уникальный серийный номер отзываемого сертификата. Данное значение потребуется вам, если вы захотите просмотреть список отзыва и убедиться в наличии в нем сертификата, как описано в последнем шаге этого раздела.

После подтверждения действия ЦС выполнит отзыв сертификата. Однако удаленные системы, использующие ЦС, не имеют возможности проверить отзыв сертификатов. Пользователи и серверы смогут использовать этот сертификат, пока список отзыва сертификатов ЦС (CRL) не будет распространен по всем системам, использующим данный ЦС.

На следующем шаге мы сгенерируем CRL или обновим существующий файл `crl.pem`.

???????????????? ?????? ??????? ????????????????

Мы отозвали сертификат, и теперь нам нужно обновить список отозванных сертификатов на сервере ЦС. После получения обновленного списка отзыва вы сможете определить, какие пользователи и системы имеют действующие сертификаты в вашем ЦС.

Чтобы сгенерировать CRL, запустите команду `easy-rsa` с опцией `gen-crl`, оставаясь в директории `~/easy-rsa`:

```
./easyrsa gen-crl
```

Если вы использовали фразу-пароль при создании файла `ca.key`, вам будет предложено ввести ее. Команда `gen-crl` сгенерирует файл с именем `crl.pem`, содержащий обновленный список отозванных сертификатов для этого ЦС.

Далее вам нужно будет передавать обновленный файл `crl.pem` на все серверы и клиенты, использующие этот ЦС, при каждом запуске команды `gen-crl`. В противном случае клиенты и системы сохранят доступ к сервисам и системам, использующим ваш ЦС, так как данным сервисам нужно сообщить об отзыве сертификата.

?????????? ??????? ??????? ????????????????

Мы сгенерировали список CRL на сервере ЦС, и теперь нам нужно передать его на удаленные системы, использующие ваш ЦС. Для передачи этого файла на ваши серверы можно использовать команду `scp`.

В этом обучающем руководстве описывается генерирование и распространение списка CRL вручную. Хотя существуют более надежные автоматические методы распространения и проверки списков отзыва (например [OCSP-Stapling](#)), настройка этих методов не входит в состав данного обучающего руководства.

Войдите на сервер ЦС как пользователь без прав root и запустите следующую команду, указав IP-адрес или имя DNS вашего сервера вместо `your_server_ip`:

```
scp ~/easy-rsa/pki/crl.pem sammy@your_server_ip:/tmp
```

Теперь файл передан на удаленную систему и нужно только отправить новую копию списка отзыва во все сервисы.

---

Revision #2

Created 8 January 2023 19:51:36 by Scoot

Updated 8 January 2023 20:39:54 by Scoot